Rateless Resilient Network Coding Against Byzantine Adversaries

Wentao Huang, Tracey Ho, Hongyi Yao
California Institute of Technology, USA
{whuang, tho, hyao}@caltech.edu

Sidharth Jaggi
Chinese University of Hong Kong, Hong Kong
jaggi@ie.cuhk.edu.hk

Abstract—This paper considers rateless network error correction codes for reliable multicast in the presence of adversarial errors. Most existing network error correction codes are designed for a given network capacity and maximum number of errors known a priori to the encoder and decoder. However, in certain practical settings it may be necessary to operate without such a priori knowledge. We present rateless coding schemes for two adversarial models, where the source sends more redundancy over time, until decoding succeeds. The first model assumes there is a secret channel between the source and the destination that the adversaries cannot overhear. The rate of the channel is negligible compared to the main network. In the second model, instead of a secret channel, the source and destination share random secrets independent of the input information. The amount of secret information required is negligible compared to the amount of information sent. Both schemes are optimal in that decoding succeeds with high probability when the total amount of information received by the sink satisfies the cut set bound with respect to the amount of message and error information. The schemes are distributed, polynomial-time and end-to-end in that other than the source and destination nodes, other intermediate nodes carry out classical random linear network coding.

I. INTRODUCTION

Network coding is a technique that allows the mixing of data at intermediate network nodes instead of simply relaying them. It has been shown theoretically that such codes are capable of achieving multicast network capacity and can be implemented in a distributed manner, as well as improving robustness against packet losses and link failures [1], [2], [3], [4]. However, comparing with pure forwarding of packets, network coding is more vulnerable to attack by malicious adversaries that inject corrupted packets, since a single corrupted packet is mixed with other packets in the network. The use of coding to correct such errors information theoretically was introduced by [5], [6], and capacity-achieving network error correction codes have been proposed for various adversary and network models, e.g. [7], [8], [9]. However, most existing schemes assume a given min cut (capacity) of the network and maximum number of adversarial errors for the purposes of code design and encoding. But such an assumption may be overly restrictive in many practical settings. For example, in large peer-to-peer content distribution networks, estimating network capacity is not easy, and the capacity is likely to change over time as users join and leave the network. Furthermore, it would be even more difficult to decide the number of malicious nodes and their strength. This issue becomes more serious if the source is multicasting to many destinations, where different destinations may require different code constructions to suit their own parameters.

This paper proposes rateless network error correction codes that do not require an a priori estimates of the network capacity and number of errors. The source transmits redundancy incrementally until decoding succeeds. The supply of encoded packets is potentially limitless and the number of encoded packets actually transmitted is determined by the number of errors that occur. A number of related works e.g. [10], [11], [12] propose cryptographic schemes that can be used to detect and remove errors in rateless network codes, while [13] proposes a rateless network error correction scheme that requires cryptographic means of verifying successful decoding. In contrast, our work presents the first completely information-theoretic rateless network error correction codes.

We design two algorithms targeting different network models. In the first model, also studied in [8], there is a secret channel between the source and the destination that is hidden from the adversary (who is omniscient except for the secret), and the rate of the channel is negligible compared to the network. In this case over time we incrementally send more linearly dependent redundancy of the source message through the network to combat erasures, and incrementally send more (linearly independent) short hashes of the message on the secret channel to eliminate fake information. The destination amasses both kinds of redundancy until he decodes successfully. The code will adapt to the actual min cut of the network as well as the number of errors.

The second scenario is the random secret model [9], where instead of a secret channel, the source and destination share a "small" fixed random secret that is independent of the input message. The amount of secrets required is again negligible compared to the amount of information sent. The random secret model may be more realistic than the secret channel model because it allows the source and destination to share their secrets in advance and use them for later communication over time. It is also possible for source and destination to share only a secret seed and generate pseudo random sequences with the seed [14]. Compared to the secret channel model, the challenge is that both linearly dependent and independent redundancy must be sent over the public and unreliable network. Again, our code will adapt to the network and adversary parameters.

Both schemes are distributed with polynomial-time com-

plexity of design and implementation. They assume no knowledge of the topology and work in both wired and wireless networks. Moreover, implementation involves only slightly modifying the source encoder and destination decoder, while internal nodes use standard random linear network coding.

II. NETWORK MODELS

A. Adversary Model

The source Alice wishes to communicate reliably with the destination Bob over a general network, where there is a hidden attacker Calvin who wants to disrupt the communication. Calvin is assumed to be able to observe all the transmissions over the network, and know the encoding and decoding schemes at Alice, Bob, as well as all other intermediate nodes. He is also aware of the network topology. Calvin can corrupt transmitted packets or inject erroneous packets. Finally, we assume Calvin to be computationally unbounded, so information-theoretic security is required in this case.

However, we assume that Calvin's knowledge is limited in some aspects. Two limitation models are discussed in this paper. For the first model, in addition to the given network, we assume there is a secret channel between Alice and Bob, *i.e.*, the information transmitted on this channel will not be observed or modified by Calvin [8]. However, the rate of the channel is negligible compared to the network. In the second model, we assume the source and destination share a small amount of random secret information that is independent with the input information [9]. Again, the amount of secret information required is negligible compared to the amount of information sent. As we will show later, the differences between the two models and the respective code constructions are substantial.

B. Network Model

We model the network in the general case as a hypergraph where nodes are vertices and hyperedges are directed from the transmitting nodes to the set of the receiving nodes [15]. Let $\mathcal{E}$ be the set of hyperedges and $\mathcal{T}$ be the set of nodes. Alice and Bob are not assumed to have knowledge of the topology of the hypergraph. They also may not know the capacity of the network as well as the number of errors that the adversary can inject.

Source Alice encodes her information bits into a batch of b packets by the encoding schemes described in subsequent sections. Each packet contains a sequence of $n + b$ symbols from the finite field $\mathbb{F}_q$. Let matrix $X_0 = \mathbb{F}_q^{b \times (n+b)}$ be one batch of packets from Alice that is desired to be communicated to Bob. We call the successful communication of one batch of information bits X_0 to the destination a session. For clarity, we focus on one such session. In the rateless setting, because Alice does not know the network capacity and error patterns, a session may require multiple network transmissions until Bob receives enough redundancy to cope with errors and decode correctly. Assume in general that a session involves N stages, *i.e.*, N uses of the network, where N is a variable. During the i-th stage, $1 \le i \le N$, denote the capacity (min cut from Alice to Bob) of the network as M_i, and the number of errors (min cut from Calvin to Bob) that the adversary injects as z_i. We assume $z_i < M_i$, otherwise the network capacity is completely filled with errors and it is not possible to transmit anything. For any realistic network, M_i is always bounded. For example, let c_i be the number of transmission opportunities that occur to the source during the i-th stage, then $M_i \le c_i$. For convenience we further assume $c_i \le \bar{c}$, $\forall i$.

III. CODE CONSTRUCTION FOR SECRET CHANNEL MODEL

A. Encoder

Alice's encoder has a structure similar to the secret channel model in [8], but operates in a rateless manner. In each session Alice transmits nb incompressible information symbols from $\mathbb{F}_q$ to Bob. Alice arranges them into a matrix $W \in \mathbb{F}_q^{b \times n}$, and encodes $X_0 = (W \ I_b)$, where I_b is the identity matrix of dimension b. Then, as in [16], [4], Alice performs random linear combinations to the rows of X_0 to generate her transmitted packets. Specifically, Alice draws a random matrix $K_1 \in \mathbb{F}_q^{c_1 \times b}$ and encodes $X_1 = K_1 X_0$. Note that the redundant identity matrix receives the same linear transform so we can recover X_0. X_1 is then send over a network where intermediate nodes implement random linear coding. In addition, Alice will hash the message and send it through the secret channel. She sets $\alpha_1 = bc_1$, and draws random symbols $r_1, ..., r_{\alpha_1+1}$ independently and uniformly from $\mathbb{F}_q$. Note that the $\{r_j\}$ are drawn secretly so that Calvin cannot observe them. Let $D_1 = [d_{kj}] \in \mathbb{F}_q^{(n+b) \times (\alpha_1+1)}$, where $d_{kj} = (r_j)^k$, and then the hash is computed as $H_1 = X_0 D_1$. Finally Alice sends $r_1, ..., r_{\alpha_1+1}$ and H_1 to Bob through the secret channel. The size of the secret is $(\alpha_1+1)(b+1)$, which is asymptotically negligible in n.

Alice then keeps sending more redundant information to Bob as follows. For the i-th stage, $i \ge 2$, Alice draws a random matrix $K_i \in \mathbb{F}_q^{c_i \times b}$, encodes $X_i = K_i X_0$, and sends X_i over the network. In addition, Alice again draws $r_1, ..., r_{\alpha_i}$ randomly from $\mathbb{F}_q$ secretly, where $\alpha_i = bc_i$. She then constructs $D_i = [d_{kj}] \in \mathbb{F}_q^{(n+b) \times \alpha_i}$, $d_{kj} = (r_j)^k$, and computes $H_i = X_0 D_i$. Alice eventually sends $r_1, ..., r_{\alpha_i}$ and H_i to Bob through the secret channel. The size of the secret is $\alpha_i(b+1)$, again asymptotically negligible in n. Alice repeats this procedure until Bob indicates decoding success. If a success is indicated, Alice ends the current session and moves onto the next session.

B. Decoder

The network performs a classical distributed network code (which is shown to suffice to achieve capacity for multicast [4]). Specifically, each packet transmitted by an intermediate node is a random linear combination of its incoming packets. For the i-th stage, we can describe this linear relation as

$$Y_i = [T_i \ Q_i] \begin{bmatrix} X_i \\ Z_i \end{bmatrix},$$

where $Y_i \in \mathbb{F}_q^{M_i \times (n+b)}$ is Bob's received observation, $Z_i \in \mathbb{F}_q^{z_i \times (n+b)}$ is the errors injected by Calvin, and T_i and Q_i are

defined to be the transfer matrix from Alice to Bob and from Calvin to Bob, respectively. By stacking all the batches of observations received by the i-th stage, let

$$Y^{(i)} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_i \end{bmatrix}, \quad Z^{(i)} = \begin{bmatrix} Z_1 \\ \vdots \\ Z_i \end{bmatrix},$$

$$H^{(i)} = [H_1 \ ... \ H_i], \quad D^{(i)} = [D_1 ... D_i],$$

$$\hat{T}^{(i)} = \begin{bmatrix} T_1 K_1 & Q_1 & 0 & ... & 0 \\ T_2 K_2 & 0 & Q_2 & ... & 0 \\ \vdots & \vdots & & & \\ T_i K_i & 0 & 0 & ... & Q_i \end{bmatrix} = \begin{bmatrix} T^{(i)} \mid Q^{(i)} \end{bmatrix}.$$

Then we have

$$Y^{(i)} = \begin{bmatrix} T^{(i)} & Q^{(i)} \end{bmatrix} \begin{bmatrix} X_0 \\ Z^{(i)} \end{bmatrix}, \tag{1}$$

$$X_0 D^{(i)} = H^{(i)}, \tag{2}$$

where (1) follows from the network transform, and (2) follows from the code construction. Notice that only $Y^{(i)}$, $D^{(i)}$ and $H^{(i)}$ are available to Bob, and he needs to recover X_0 from equations (1), (2). Bob can accomplish this only if X_0 is in the row space of $Y^{(i)}$.

Suppose X_0 indeed lies in the row space of $Y^{(i)}$ (it happens with high probability for some i as shown later), then there exists X^s such that

$$X_0 = X^s Y^{(i)}. \tag{3}$$

Therefore Bob only needs to find X^s, which may be achieved by solving

$$X^s Y^{(i)} D^{(i)} = H^{(i)}. \tag{4}$$

If (4) has a unique solution for X^s, Bob reconstructs X_0 according to (3) and feedbacks an acknowledgement of decoding success to Alice. If there exists no solution for (4), Bob waits for more redundancy to come. Otherwise, if there are multiple solutions for (4), Bob declares a decoding failure.

C. Performance

In the following we will show that the probability of error, including the events that Bob declares an error, or the events that X_0 is not in the row space of $Y^{(i)}$, or there exists some other $X' \neq X_0$ that satisfies (4), is vanishing as $q \to \infty$.

We first show that under proper conditions, X_0 is in the row span of $Y^{(i)}$. The following lemma is well-known [17]:

Lemma 1: If the linear transform $\hat{T}^{(i)}$ has full column rank, *i.e.*, $\text{Rank}(\hat{T}^{(i)}) = b + \sum_{j=1}^{i} z_j$, then it is left-invertible, and there exists X^s such that $X^s Y^{(i)} = X_0$.

Now we show that $\hat{T}^{(i)}$ almost always has either full column rank or full row rank as $q \to \infty$.

Lemma 2: If $b + \sum_{j=1}^{i} z_j \leq \sum_{j=1}^{i} M_j$, then $\hat{T}^{(i)}$ has full column rank with high probability[1].

---

[1] Event E happens with high probability ($w.h.p.$) if $\lim_{q\to\infty} \Pr\{E\} = 1$.

Proof: The proof follows an idea similar to [8] with the difference that we consider communications over multiple stages. Notice that $\hat{T}^{(i)} = [T^{(i)} \ Q^{(i)}]$. Since $b + \sum_{j=1}^{i} z_j \leq \sum_{j=1}^{i} M_j$, it follows $b \leq \sum_{j=1}^{i} M_j$. Because T_j has full rank and K_j are random matrices, if $b \leq \sum_{j=1}^{i} M_j$, then the probability that the columns of $T^{(i)}$ are linearly dependent is upper bounded by $b/q \to 0$ by the Schwartz-Zippel lemma. So $T^{(i)}$ has full column rank. Without loss of generality we assume $Q^{(i)}$ also has full column rank, otherwise we can select a basis of the column space of $Q^{(i)}$ and reformulate the problem with a reduced z_i. Furthermore, by [18], if $b + \sum_{j=1}^{i} z_j \leq \sum_{j=1}^{i} M_j$, the probability that the column spans of $T^{(i)}$ and $Q^{(i)}$ intersects anywhere other than in the zero vector is upper bounded by $i^2 |\mathcal{T}||\mathcal{E}|q^{-1}$ for a fix adversary pattern. Since Calvin can choose his locations in at most $\binom{i\lceil\mathcal{E}\rceil}{\sum_j z_j}$ ways, by the union bound, the probability that $T^{(i)}$ and $Q^{(i)}$ intersects is bounded by $\binom{i|\mathcal{E}|}{\sum_j z_j} i^2 |\mathcal{T}||\mathcal{E}|q^{-1} \to 0$. Hence $\hat{T}^{(i)}$ has full column rank with high probability. ∎

Corollary 1: If $b + \sum_{j=1}^{i} z_j \leq \sum_{j=1}^{i} M_j$, then with high probability there exists a X^s such that $X^s Y^{(i)} = X_0$.

Next we need to show the solution is unique, *i.e.*, the hash is strong enough so that Bob can distill the injected error.

Lemma 3: For any $X' \neq X_0$, the probability that $X'D^{(i)} = H^{(i)}$ is bounded from above by $((n+b)/q)^{\sum_{k=1}^{i} \alpha_k + 1}$.

Proof: It is equivalent to consider the probability that $(X' - X_0)D^{(i)} = 0$. Since $X' - X_0 \neq 0$, there is at least one row in which X' differs from X_0. Denote this row of $X' - X_0$ as $(x_1, ..., x_{n+b})$, then the j-th entries of the corresponding row of $(X' - X_0)D^{(i)}$ is $F(r_j) = \sum_{k=1}^{n+b} x_k r_j^k$. Because $F(r_j)$ is not the zero polynomial, the probability (over r_j) that $F(r_j) = 0$ is at most $(n+b)/q$. Because $D^{(i)}$ has $\sum_{k=1}^{i} \alpha_k + 1$ columns, and all r_j, $1 \leq j \leq \sum_{k=1}^{i} \alpha_k + 1$, are independently chosen, the probability that the entire row is a zero vector is at most $((n+b)/q)^{\sum_{k=1}^{i} \alpha_k + 1}$. This is an upper bound of the probability that the entire matrix of $(X' - X_0)D^{(i)}$ is zero. ∎

Lemma 4: The probability that there exists $V^s \neq X^s$ such that $V^s Y^{(i)} \neq X_0$ but $V^s Y^{(i)} D^{(i)} = H^{(i)}$ is upper bounded by $(n+b)^{\sum_{k=1}^{i} \alpha_k + 1}/q \to 0$.

Proof: Note that the dimension of V^s is

$$b \sum_{j=1}^{i} M_j \leq b \sum_{j=1}^{i} c_j$$

over $\mathbb{F}_q$. So by invoking Lemma 3 and then take the union bound over all possible choices of V^s, the claim follows. ∎

Now we are ready to present the main result for the shared secret model.

Theorem 1: $\forall i$ such that $b + \sum_{j=1}^{i} z_j \leq \sum_{j=1}^{i} M_j$, with the proposed coding scheme, Bob is able to decode X_0 correctly with high probability at the i-th stage. Otherwise, Bob waits for more redundancy instead of decoding erroneous packets.

Proof: By Corollary 1, we can solve X_0 from (3) and (4) if $b + \sum_{j=1}^{i} z_j \leq \sum_{j=1}^{i} M_j$. By Lemma 4, if a solution exists, it is correct and unique. Otherwise, there is no solution to (4) and by the algorithm Bob waits for more redundancy. ∎

Theorem 1 shows the code is optimal in that decoding succeeds with high probability whenever the total amount of information received by the sink satisfies the cut set bound, *i.e.*, $b + \sum_{j=1}^{i} z_j \leq \sum_{j=1}^{i} M_j$. If the bound is not satisfied, then it is not possible for Bob to decode correctly under any coding scheme. The following result shows that the code is rate optimal if the network capacity and number of errors are i.i.d. across stages.

Theorem 2: Assume M_i, z_i, $i = 1, 2...$ are i.i.d. random variables with mean $\mathbb{E}[M]$ and $\mathbb{E}[z]$, respectively. If there exists $\epsilon > 0$ such that $\mathbb{E}[M] - \mathbb{E}[z] \geq \epsilon$, then with the proposed coding scheme, Bob is able to decode X_0 correctly with high probability in a finite number of stages. Further, on average the code achieves rate

$$r \geq \frac{b}{b + \bar{c} - 1} \left(\mathbb{E}[M] - \mathbb{E}[z] \right).$$

Proof: Let $L = b/\epsilon$, and let random variable $\Delta_j = M_j - z_j$, so $\mathbb{E}[\Delta_j] = \epsilon$ and denote $\text{Var}[\Delta_j] = \sigma_\Delta^2 < \infty$. Then for $N \geq L$, $\mathbb{E}[\sum_{j=1}^{N} \Delta_j] = N\epsilon = b + (N - L)\epsilon$. By Chebyshev's inequality,

$$\Pr\left\{ \sum_{j=1}^{N} \Delta_j \geq b \right\}$$

$$= \Pr\left\{ \sum_{j=1}^{N} \Delta_j \geq \mathbb{E}\left[\sum_{j=1}^{N} \Delta_j \right] - (N - L)\epsilon \right\}$$

$$\geq 1 - \Pr\left\{ \left| \sum_{j=1}^{N} \Delta_j - \mathbb{E}\left[\sum_{j=1}^{N} \Delta_j \right] \right| \geq (N - L)\epsilon \right\}$$

$$\geq 1 - \frac{N\sigma_\Delta^2}{(N - L)^2 \epsilon^2} \to 1 \quad \text{as } N \to \infty.$$

So with high probability there exists finite $N \geq L$ such that $b + \sum_{j=1}^{N} z_j \leq \sum_{j=1}^{N} M_j$. And by Theorem 1, Bob is able to decode successfully at stage N. Next we determine the average rate, let

$$N = \min\left\{ i : b + \sum_{j=1}^{i} z_j \leq \sum_{j=1}^{i} M_j \right\}.$$

Then the average rate of the code is

$$r = \mathbb{E}\left[\frac{b}{N} \right] \geq \frac{b}{\mathbb{E}[N]}, \tag{5}$$

where the last inequality follows from Jensen's inequality. Denote $S_\tau = \sum_{j=1}^{\tau} M_j - \sum_{j=1}^{\tau} z_j$, then N is a stopping time for the random process S_τ. Therefore, by the first Wald identity [19],

$$\mathbb{E}[N](\mathbb{E}[M] - \mathbb{E}[z]) = \mathbb{E}[S_N]$$
$$\leq b + \bar{c} - 1.$$

Therefore

$$\mathbb{E}[N] \leq \frac{b + \bar{c} - 1}{\mathbb{E}[M] - \mathbb{E}[z]}.$$

Substituting into (5), it follows

$$r \geq \frac{b}{b + \bar{c} - 1} \left(\mathbb{E}[M] - \mathbb{E}[z] \right).$$

∎

Notice that as we choose a sufficiently large b, then the rate of the code is approaching $\mathbb{E}[M] - \mathbb{E}[z]$, which is shown to be the maximal achievable rate for networks with Byzantine adversaries [20]. The computational cost of design, encoding, and decoding is dominated by the cost of carrying out the matrix multiplication $Y^{(i)} D^{(i)}$ in (4), which is $O(n(i\bar{c})^3)$. Notice that because $Y^{(i)}$ and $D^{(i)}$ grow regularly, $Y^{(i)} D^{(i)}$ is a block of $Y^{(i+1)} D^{(i+1)}$, therefore careful implementation of the algorithm can improve complexity (though not in the order sense) by building on the results from the last stage. Assume in general that $Y^{(i)} D^{(i)} = A^T$ and

$$\left(Y^{(i+1)} D^{(i+1)} \right)^T = \begin{bmatrix} A & C \\ B & D \end{bmatrix}.$$

Then at stage $i+1$ we only need to perform the multiplications corresponding to blocks B, C, and D. The same trick applies when we are to perform row reduction on $Y^{(i)} D^{(i)}$. Suppose at the i-th stage we have already reduced A into row echelon form with matrix R, *i.e.*, with high probability it follows (otherwise there is a decoding error)

$$RA = A' = \begin{bmatrix} I \\ \mathbf{0} \end{bmatrix}.$$

At the $(i + 1)$-th stage we want to reduce $\left(Y^{(i+1)} D^{(i+1)} \right)^T$ into row echelon form based on the knowledge of R. We can construct the row operations as the following steps: i) multiply R to reduce block A and obtain $[A' \ RC]$ in the upper blocks; ii) use A' to cancel block B to zero; iii) perform row reduction on the lower right block corresponding to D; iv) use the row reduced lower right block to cancel the upper right block to zero. Formally, let

$$R' = \begin{bmatrix} I & -RC|\mathbf{0} \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & D^- \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ -B|\mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} R & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix}$$

where D^- is the matrix that row reduces $[B|\mathbf{0}]RC + D$, where the $\mathbf{0}$ after B is defined as a zero padding sub-matrix of appropriate dimension. Then it follows

$$R' Y^{(i+1)} D^{(i+1)} = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \\ & I \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Finally we only need to permute the rows to place the identity matrix on top. Note that by this algorithm at every stage we only need to perform row reduction on a small block (corresponding to D) plus several multiplications.

IV. CODE CONSTRUCTION FOR RANDOM SECRET MODEL

In this section we consider the case that a secret channel is not available between Alice and Bob, instead they only share a "small" random secrets whose size is negligible compared to

the amount of information sent. The random secret model is similar to the previously discussed secret channel model with the difference that the secret information should be random and independent with the source message X_0. Therefore in this case Alice cannot send the hash of X_0 to Bob secretly and reliably. Nutman and Langberg [9] modifies the scheme in [8] against Byzantine adversaries under the secret channel model so that it also works for the random secret model. The essential idea is to prefix X_0 with a small matrix L obtained by solving the hash relation $(L\ X_0)D = H$, where both D and H are the fixed random secrets. Then $(L\ X_0)$ is sent through the network as the new input message. However, there is no obvious way to make this scheme rateless, because in the rateless setting, the number of columns of D and H will grow over time as more linearly independent redundancy is needed. But this implies it is not possible to uniquely solve L given a fixed X_0.

The difficulty in solving L reveals the restriction of defining the hash relation in the matrix form of $X_0 D = H$. Hence a more flexible way for hashing the data is required as discussed in the following. Recall that the vectorization of a matrix is a linear transformation which converts the matrix into a column vector by stacking the columns of the matrix on top of one another. Let column vector $\boldsymbol{w} \in \mathbb{F}_q^{bn}$ be the vectorized[2] W (recall that W is the matrix of raw incompressible source data before attaching the identity matrix, defined in Section III-A).

To generate the hashes, *i.e.*, the linearly independent redundancy that is transmitted at the k-th stage. We first draw α_k symbols from the random shared secrets as $d_1^{(k)}$, $d_2^{(k)}$, ..., $d_{\alpha_k}^{(k)} \in \mathbb{F}_q$, and use them to construct the $\alpha_k \times nb$ parity check matrix $D_k = \left[d_{ij}^{(k)} \right]$, where $d_{ij}^{(k)} = \left(d_i^{(k)} \right)^j$, $\forall 1 \leq i \leq \alpha_k$, $1 \leq j \leq nb$.

Then we draw another α_k symbols from the random shared secrets as $h_1^{(k)}, h_2^{(k)}, ..., h_{\alpha_k}^{(k)}$ and use them to construct the hash of the message. Let $\boldsymbol{h}_k = (h_1^{(k)}, ..., h_{\alpha_k}^{(k)})^T$. Then the following parity check relation is enforced:

$$\begin{bmatrix} D_k & I_{\alpha_k} \end{bmatrix} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{l}_k \end{bmatrix} = \boldsymbol{h}_k, \qquad (6)$$

where I_{α_k} is the identity matrix of dimension α_k and $\boldsymbol{l}_k$ is a column vector of length α_k that can be solved uniquely from (6) as

$$\boldsymbol{l}_k = \boldsymbol{h}_k - D_k \boldsymbol{w}$$

Note that $\{\boldsymbol{l}_k\}$ does not need to be kept secret. Now we can readily construct a rateless parity check scheme based on (6):

$$\begin{bmatrix} D_1 & I_{\alpha_1} & 0 & ... & 0 \\ D_2 & 0 & I_{\alpha_2} & ... & 0 \\ \vdots & & \vdots & & \\ D_i & 0 & 0 & ... & I_{\alpha_i} \end{bmatrix} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{l}_1 \\ \vdots \\ \boldsymbol{l}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \vdots \\ \boldsymbol{h}_i \end{bmatrix}, \quad (7)$$

[2]Compared to linear transforms on the matrix W, linear transforms on the vector $\boldsymbol{w}$ are more general (every operation in the former class can be representation by an operation in the latter class, but not vice versa)

i.e., the total number of parity checks $\sum_i \alpha_i$ can grow over time if necessary.

(7) implies that we are hashing $(\boldsymbol{w}^T\ \boldsymbol{l}^T)^T$ instead of the message $\boldsymbol{w}$ itself. However, the advantage of introducing $\boldsymbol{l}_k$ is that by attaching a short suffix to $\boldsymbol{w}$, we can establish a virtual secret channel between Alice and Bob. Nevertheless, the challenge is then we need to send not only $\boldsymbol{w}$ but also $\{\boldsymbol{l}_k\}$ over the unreliable network publicly. In the following we will discuss the structure of the encoder and decoder for transmitting them successfully.

A. Encoder

In order for Bob to decode successfully, both linearly dependent redundancy and linearly independent redundancy are required. Redundant information that lies in the row space of X_0 is called linearly dependent redundancy and is used to combat erasures in the network (deletion from the row space of X_0). Other redundant information is linearly independent redundancy and is used to distill "fake information" that adversaries inject into the network (addition to the row space of X_0). In the case of the secret channel model, we send linearly dependent redundancy $\{X_i\}$ in the network and send linearly independent redundancy $\{D_i\}$ and $\{H_i\}$ on the secret channel. Compared to that, in the case of the random secret model, a secret channel is not available and both linearly dependent and independent redundancy must be sent through the public and unreliable network.

However, notice that the linearly dependent redundancy corresponds to long messages that are usually arranged into long packets, while the linearly independent redundancy is short. Its size is chosen to be independent with n, as it is desired that the amount of random secrets required is negligible comparing to the amount of information sent. Therefore, it is convenient to encode and send the two kinds of redundancy separately (for example, it would be wasteful of resources if linearly independent redundancy is sent in normal packets because it is too short to fill a packet) as long packets and short packets, respectively. We define M_i, z_i, c_i, $\bar{c}$ for long packets as described in Section II. For short packets, denote $\bar{M}_i$, $\bar{c}_i$ and $\bar{z}_i$ as the min cut from Alice to Bob, the number of available transmission opportunities, and the min cut from Calvin to Bob at stage i, respectively. Similarly we assume $\bar{z}_i < \bar{M}_i$, $\forall i$.

We first discuss the encoding scheme for linearly dependent redundancy. The source input message is arranged as a $b \times n$ matrix W. Then we encode $X_0 = (W\ I_b)$. At the i-th stage, Alice draws a random matrix $K_i \in \mathbb{F}_q^{c_i \times b}$, and encodes the long packets $X_i = K_i X_0$.

To generate the linearly independent redundancy, at stage i Alice sets $\alpha_i = i\sigma m$ (the choices of σ and m are discussed in the next paragraph), solves $\boldsymbol{l_i}$ according to (6), and arranges the column vector into a $\sigma \times im$ matrix $\mathcal{L}_i$. Then she let $L_j = (\mathcal{L}_j \quad \boldsymbol{0}_D \quad \boldsymbol{0}_j \quad I_\sigma)$, $1 \leq j \leq i$, where $\boldsymbol{0}_D$ is a zero matrix of size $\sigma \times (i-j)m$, and $\boldsymbol{0}_j$ is the zero matrix of size $\sigma \times (j-1)\sigma$. $\boldsymbol{0}_D$ is dummy and is used to align $\mathcal{L}$, and $\boldsymbol{0}_j$ is used to align the identity matrix. Alice then draws a random

matrix G_i of size $\bar{c}_i \times i\sigma$ and encodes the short packets as

$$A_i = G_i \begin{bmatrix} \begin{array}{cccc} L_1 & 0 & \dots & 0 \\ \hline L_2 & & \dots & 0 \\ \hline & \dots & & \\ \hline & & L_i & \end{array} \end{bmatrix} = G_i L^{(i)}.$$

In order to eliminate the "fake" information injected by the adversaries, Alice should introduce an adequate amount of linearly independent redundancy, *i.e.*, choosing $\{\alpha_i\}$ appropriately. Alice may choose any σ such that $\sigma \le \bar{M}_i - \bar{z}_i, \ \forall i$ (e.g., $\sigma = 1$ is a safe choice). She then chooses m such that $\sigma m \ge 2b\bar{c} + 2\sigma\bar{c} + 1$. Note that the size of the secret, $i(i+1)\sigma m/2$, is again negligible in n.

Finally, at the i-th stage Alice sends X_i as long packets with packet length $n + b$, and A_i as short packets with packet length $i(m + \sigma)$. Alice repeats this procedure until Bob decodes successfully.

B. Decoder

At the i-th stage, Bob receives long packets Y_i and short packets J_i:

$$Y_i = T_i X_i + Q_i Z_i, \qquad (8)$$
$$J_i = \bar{T}_i A_i + \bar{Q}_i E_i, \qquad (9)$$

where $T_i \in \mathbb{F}_q^{M_i \times c_i}$, $\bar{T}_i \in \mathbb{F}_q^{\bar{M}_i \times \bar{c}_i}$ are the transfer matrices between Alice and Bob, $Q_i \in \mathbb{F}_q^{M_i \times z_i}$, $\bar{Q}_i \in \mathbb{F}_q^{\bar{M}_i \times \bar{z}_i}$ are the transfer matrices between Calvin and Bob, and $Z_i \in \mathbb{F}_q^{z_i \times (n+b)}$, $E_i \in \mathbb{F}_q^{\bar{z}_i \times i(m+\sigma)}$ are the errors injected to long packets and short packets, respectively.

Bob then stacks the long and short packets that he has received so far to get

$$Y^{(i)} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_i \end{bmatrix}, \quad J^{(i)} = \begin{bmatrix} \begin{array}{cccc} J_1 & 0 & \dots & 0 \\ \hline J_2 & & \dots & 0 \\ \hline & \dots & & \\ \hline & & J_i & \end{array} \end{bmatrix},$$

where dummy matrix $\mathbf{0}_D$ is padded to $\{J_k\}$ in the same way as it is padded to $\{L_k\}$.

Bob evaluates the rank of $Y^{(i)}$, and waits to receive more packets if $r_i = \text{Rank}(Y^{(i)}) < b$. When $r_i \ge b$, Bob tries to decode. Without loss of generality we assume the rows of $Y^{(i)}$ are linearly independent. Otherwise, Bob selects r_i linearly independent rows from $Y^{(i)}$ and proceeds similarly. He then picks a basis for the column space of $Y^{(i)}$. As will be shown later, the last b columns of $Y^{(i)}$ (corresponding to the identity matrix in X_0) are linearly independent *w.h.p.*, so they are chosen and is denoted by a $r_i \times b$ matrix $\hat{T}^{(i)}$. Without loss of generality (by permuting the columns if necessary) we assume that the remaining $r_i - b$ linearly independent columns correspond to the first $r_i - b$ columns of $Y^{(i)}$, denoted by a $r_i \times (r_i - b)$ matrix $T''^{(i)}$. So we can expand $Y^{(i)}$ with respect to this basis as

$$Y^{(i)} = [T''^{(i)} \ \hat{T}^{(i)}] \begin{bmatrix} I_{r_i - b} & F^Z & 0 \\ 0 & F^X & I_b \end{bmatrix}, \qquad (10)$$

where F^Z and F^X are matrices of coefficients.

Bob deals with $J^{(i)}$ in a similar way. Let $\bar{r}_i$ be the rank of $J^{(i)}$, $\hat{\bar{T}}^{(i)} \in \mathbb{F}_q^{\bar{r}_i \times i\sigma}$ be the last $i\sigma$ columns of $J^{(i)}$, and $\bar{T}''^{(i)} \in \mathbb{F}_q^{\bar{r}_i \times (\bar{r}_i - i\sigma)}$ be the first $\bar{r}_i - i\sigma$ columns of $J^{(i)}$. Then *w.h.p.* $[\bar{T}''^{(i)} \ \hat{\bar{T}}^{(i)}]$ consists a basis of the column space of $J^{(i)}$, and we can write

$$J^{(i)} = [\bar{T}''^{(i)} \ \hat{\bar{T}}^{(i)}] \begin{bmatrix} I_{\bar{r}_i - i\sigma} & F^E & 0 \\ 0 & F^A & I_{i\sigma} \end{bmatrix}, \qquad (11)$$

where F^E and F^A are matrices of coefficients.

Equations (10) and (11) characterize the relationship between the received observations and the input messages due to the effect of the network transform. In order to decode successfully, Bob needs to take into account the built-in redundancy of the message, *i.e.*, the relation between $\boldsymbol{w}$ and $\{\boldsymbol{l}_i\}$, as follows. $\forall i$, split X_0 and $L^{(i)}$ as:

$$X_0 = [X_a^{(i)} \ X_b^{(i)} \ X_c^{(i)}], \qquad (12)$$
$$L^{(i)} = [L_a^{(i)} \ L_b^{(i)} \ L_c^{(i)}], \qquad (13)$$

where $X_a^{(i)}$ are the first $r_i - b$ columns of X_0, $X_c^{(i)}$ are the last b columns of X_0, and $X_b^{(i)}$ are the remaining columns in the middle; $L_a^{(i)}$ are the first $\bar{r}_i - i\sigma$ columns of $L^{(i)}$, $L_c^{(i)}$ are the last $i\sigma$ columns of $L^{(i)}$, and $L_b^{(i)}$ are the remaining columns in the middle. Let $\boldsymbol{x}_a^{(i)}$, $\boldsymbol{x}_b^{(i)}$ and $\boldsymbol{x}_c^{(i)}$ be the vectorized versions of $X_a^{(i)}$, $X_b^{(i)}$ and $X_c^{(i)}$. Let $\boldsymbol{l}_a^{(i)}$, $\boldsymbol{l}_b^{(i)}$ and $\boldsymbol{l}_c^{(i)}$ be the vectorized versions of $L_a^{(i)}$, $L_b^{(i)}$ and $L_c^{(i)}$ omitting the dummy $\mathbf{0}_D$. By construction it follows that,

$$\begin{bmatrix} \boldsymbol{x}_a^{(i)} \\ \hline \boldsymbol{x}_b^{(i)} \\ \hline \boldsymbol{l}_a^{(i)} \\ \hline \boldsymbol{l}_b^{(i)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w} \\ \hline \boldsymbol{l}_1 \\ \vdots \\ \boldsymbol{l}_i \end{bmatrix}. \qquad (14)$$

Note that $\boldsymbol{x}_c^{(i)}$ and $\boldsymbol{l}_c^{(i)}$ are left out because they correspond to the redundant identity matrix. Now Bob constructs two matrix B_{top} and B_{mid} as defined in (15) and (16), respectively. Here $f_{i,j}^Z$ and $f_{i,j}^E$ are the $(i,j)^{th}$ entries of matrix F^Z and F^E, respectively, and $\beta = n + b - r_i$, $\gamma = i(m + \sigma) - \bar{r}_i$. He then deletes all columns in B_{mid} corresponding to the positions of the dummy zero padding when vectoring $L^{(i)}$, and obtain a submatrix B'_{mid}. Finally, Bob let

$$B_{bot} = \begin{bmatrix} D_1 & I_{\alpha_1} & 0 & \dots & 0 \\ D_2 & 0 & I_{\alpha_2} & \dots & 0 \\ \vdots & & \vdots & & \\ D_i & 0 & 0 & \dots & I_{\alpha_i} \end{bmatrix},$$

Notice that if Bob permutes the columns of $Y^{(i)}$ and $J^{(i)}$ when constructing $T''^{(i)}$ and $\bar{T}''^{(i)}$, then he needs to permute the columns of B_{bot} accordingly. Then he tries to solve the

$$B_{top} = \begin{bmatrix} -f^Z_{1,1}\hat{T}^{(i)} & -f^Z_{2,1}\hat{T}^{(i)} & ... & -f^Z_{r_i-b,1}\hat{T}^{(i)} & \hat{T}^{(i)} & 0 & ... & 0 \\ -f^Z_{1,2}\hat{T}^{(i)} & -f^Z_{2,2}\hat{T}^{(i)} & ... & -f^Z_{r_i-b,2}\hat{T}^{(i)} & 0 & \hat{T}^{(i)} & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -f^Z_{1,\beta}\hat{T}^{(i)} & -f^Z_{2,\beta}\hat{T}^{(i)} & ... & -f^Z_{r_i-b,\beta}\hat{T}^{(i)} & 0 & 0 & ... & \hat{T}^{(i)} \end{bmatrix} \tag{15}$$

$$B_{mid} = \begin{bmatrix} -f^E_{1,1}\hat{\bar{T}}^{(i)} & -f^E_{2,1}\hat{\bar{T}}^{(i)} & ... & -f^E_{\bar{r}_i-i\sigma,1}\hat{\bar{T}}^{(i)} & \hat{\bar{T}}^{(i)} & 0 & ... & 0 \\ -f^E_{1,2}\hat{\bar{T}}^{(i)} & -f^E_{2,2}\hat{\bar{T}}^{(i)} & ... & -f^E_{\bar{r}_i-i\sigma,2}\hat{\bar{T}}^{(i)} & 0 & \hat{\bar{T}}^{(i)} & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -f^E_{1,\gamma}\hat{\bar{T}}^{(i)} & -f^E_{2,\gamma}\hat{\bar{T}}^{(i)} & ... & -f^E_{\bar{r}_i-i\sigma,\gamma}\hat{\bar{T}}^{(i)} & 0 & 0 & ... & \hat{\bar{T}}^{(i)} \end{bmatrix} \tag{16}$$

equations:

$$B \begin{bmatrix} \boldsymbol{x}_a^{(i)} \\ \boldsymbol{x}_b^{(i)} \\ \boldsymbol{l}_a^{(i)} \\ \boldsymbol{l}_b^{(i)} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{T}}^{(i)}\boldsymbol{f}^X \\ \hat{\bar{\boldsymbol{T}}}^{(i)}\boldsymbol{f}^A \\ \boldsymbol{h_1} \\ \vdots \\ \boldsymbol{h_i} \end{bmatrix}, \tag{17}$$

where $\boldsymbol{f}^X$, $\boldsymbol{f}^A$ are the vectorized versions of F^X, F^A, respectively, $\hat{\boldsymbol{T}}^{(i)} = \mathrm{diag}[\hat{T}^{(i)}, ..., \hat{T}^{(i)}]$, $\hat{\bar{\boldsymbol{T}}}^{(i)} = \mathrm{diag}[\hat{\bar{T}}^{(i)}, ..., \hat{\bar{T}}^{(i)}]$, and the matrix B is defined as:

$$B = \begin{bmatrix} B_{top} & 0 \\ 0 & B'_{mid} \\ \hline B_{bot} \end{bmatrix}.$$

Bob tries to solve (17) and, if there exists no solution, he waits for more redundancy from Alice and tries to solve it again at the next stage. If there is a unique solution to (17), then Bob has decoded successfully with high probability. Otherwise, if there are multiple solutions, Bob declares a decoding failure.

C. Performance

In this section we show the proposed scheme will succeed with high probability and achieve the optimal rate. Our first step is to establish (10) and (11). We first consider the short packets. Note that (11) is shown by Lemma below.

Lemma 5: $\hat{\bar{T}}^{(i)}$ has full column rank with high probability.

Proof: For notational convenience we define

$$\bar{T}^{(i)} = \begin{bmatrix} \bar{T}_1 & 0 & ... & 0 \\ 0 & \bar{T}_2 & ... & 0 \\ & & ... & \\ 0 & 0 & ... & \bar{T}_i \end{bmatrix}, \ A^{(i)} = \begin{bmatrix} A_1 & 0 & ... & 0 \\ \hline A_2 & ... & 0 \\ \hline & ... & \\ \hline & A_i \end{bmatrix}$$

$$\bar{Q}^{(i)} = \begin{bmatrix} \bar{Q}_1 & 0 & ... & 0 \\ 0 & \bar{Q}_2 & ... & 0 \\ & & ... & \\ 0 & 0 & ... & \bar{Q}_i \end{bmatrix}, \ E^{(i)} = \begin{bmatrix} E_1 & 0 & ... & 0 \\ \hline E_2 & ... & 0 \\ \hline & ... & \\ \hline & E_i \end{bmatrix}$$

Then we have the concise relationship from (8) and (9):

$$\begin{aligned} J^{(i)} &= \bar{T}^{(i)}A^{(i)} + \bar{Q}^{(i)}E^{(i)} \\ &= \bar{T}^{(i)}G^{(i)}L^{(i)} + \bar{Q}^{(i)}E^{(i)}, \end{aligned} \tag{18}$$

where

$$G^{(i)} = \begin{bmatrix} G_1 & 0 & ... & 0 \\ \hline G_2 & ... & 0 \\ \hline & ... & \\ \hline & G_i \end{bmatrix}.$$

By construction, $\sum_{j=1}^{i}\bar{M}_j - \sum_{j=1}^{i}\bar{z}_j \geq i\sigma$. By an argument identical to Lemma 2, this implies that with high probability $\bar{T}^{(i)}$ and $\bar{Q}^{(i)}$ both have full column rank and span disjoint column spaces (except for the zero vector). Random matrix $G^{(i)}$ also has full rank *w.h.p.* and therefore $\bar{T}^{(i)}G^{(i)}$ and $\bar{Q}^{(i)}$ also both have full column rank and span disjoint column spaces. We can write the last $i\sigma$ columns in (18) corresponding to the redundant identity matrix in $L^{(i)}$ as

$$\hat{\bar{T}}^{(i)} = \bar{T}^{(i)}G^{(i)} + \bar{Q}^{(i)}E^{(i)}_{\text{rear}}, \tag{19}$$

where $E^{(i)}_{\text{rear}}$ are the last $i\sigma$ columns of $E^{(i)}$. Hence the columns of $\hat{\bar{T}}^{(i)}$ are linearly independent *w.h.p.*. ■

A similar argument also holds for the long packets:

Lemma 6: If $\sum_{j=1}^{i}M_j - \sum_{j=1}^{i}z_j \geq b$, then $\hat{T}^{(i)}$ has full column rank with high probability.

Proof: Consider the last b columns in (1) corresponding to the redundant identity matrix in X_0

$$\hat{T}^{(i)} = T^{(i)} + Q^{(i)}Z^{(i)}_{\text{rear}}.$$

And by Lemma (2), $T^{(i)}$ and $Q^{(i)}$ both have full column rank and span disjoint column spaces (except for the zero vector). Hence the columns of $\hat{T}^{(i)}$ are linearly independent. ■

Now we are ready to analyze the key equation (17). We first prove a related lemma.

Lemma 7: With high probability (9) and (11) are equivalent to the following equation:

$$\hat{T}^{(i)}L_b^{(i)} = \hat{\bar{T}}^{(i)}(F^A + L_a^{(i)}F^E). \tag{20}$$

Proof: We use a technique similar to [8]. Substituting (19) to (18), it follows

$$J^{(i)} = \hat{\bar{T}}^{(i)}L^{(i)} + \bar{Q}^{(i)}(E^{(i)} - E^{(i)}_{\text{rear}}L^{(i)}).$$

Then by (11) we have:

$$\begin{aligned} \hat{\bar{T}}^{(i)}L^{(i)} + \bar{Q}^{(i)}(E^{(i)} - E^{(i)}_{\text{rear}}L^{(i)}) = \\ \hat{\bar{T}}^{(i)}[0 \ F^A \ I_{i\sigma}] + \bar{T}''^{(i)}[I_{\bar{r}_i-i\sigma} \ F^E \ 0]. \end{aligned} \tag{21}$$

Therefore the columns of $\bar{T}''^{(i)}$ are spanned by the columns of $[\hat{\bar{T}}^{(i)} \ \bar{Q}^{(i)}]$. So there exists matrices V_1 and V_2 such that

$$\bar{T}''^{(i)} = \hat{\bar{T}}^{(i)} V_1 + \bar{Q}^{(i)} V_2.$$

And we can rewrite (21) as

$$\hat{\bar{T}}^{(i)} L^{(i)} + \bar{Q}^{(i)}(E^{(i)} - E_{\text{rear}}^{(i)} L^{(i)}) = \hat{\bar{T}}^{(i)}[0 \ F^A \ I_{i\sigma}] + (\hat{\bar{T}}^{(i)} V_1 + \bar{Q}^{(i)} V_2)[I_{\bar{r}_i - i\sigma} \ F^E \ 0]. \quad (22)$$

However, notice from (19) that the columns of $\hat{\bar{T}}^{(i)}$ and the columns of $\bar{Q}^{(i)}$ are $w.h.p.$ linearly independent, $i.e.$, the column spaces of $\hat{\bar{T}}^{(i)}$ and $\bar{Q}^{(i)}$ are disjoint except for the zero vector. Hence (22) implies the following set of equations:

$$\hat{\bar{T}}^{(i)} L^{(i)} = \hat{\bar{T}}^{(i)}[0 \ F^A \ I_{i\sigma}] + \hat{\bar{T}}^{(i)} V_1[I_{\bar{r}_i - i\sigma} \ F^E \ 0] \quad (23)$$
$$\bar{Q}^{(i)}(E^{(i)} - E_{\text{rear}}^{(i)} L^{(i)}) = \bar{Q}^{(i)} V_2[I_{\bar{r}_i - i\sigma} \ F^E \ 0]. \quad (24)$$

Here (23) suffices for the purpose of decoding X_0. We split (23) into three parts as in (13), and get:

$$\hat{\bar{T}}^{(i)} L_a^{(i)} = \hat{\bar{T}}^{(i)} V_1 \quad (25)$$
$$\hat{\bar{T}}^{(i)} L_b^{(i)} = \hat{\bar{T}}^{(i)} F^A + \hat{\bar{T}}^{(i)} V_1 F^E \quad (26)$$
$$\hat{\bar{T}}^{(i)} L_c^{(i)} = \hat{\bar{T}}^{(i)} \quad (27)$$

By Lemma 5, $\hat{\bar{T}}^{(i)}$ has full column rank and is left-invertible, therefore by (25) it follows $L_a^{(i)} = V_1$. Substituting it into (26)

$$\hat{\bar{T}}^{(i)} L_b^{(i)} = \hat{\bar{T}}^{(i)} F^A + \hat{\bar{T}}^{(i)} L_a^{(i)} F^E.$$

Finally, notice that by construction $L_c^{(i)}$ is the identity matrix and therefore (27) is redundant. Hence we can conclude that (9) and (11) are equivalent to

$$\hat{\bar{T}}^{(i)} L_b^{(i)} = \hat{\bar{T}}^{(i)}(F^A + L_a^{(i)} F^E)$$

. ■

By a similar argument, for long packets we have the following result:

Lemma 8: If $\sum_{j=1}^{i} M_j - \sum_{j=1}^{i} z_j \geq b$, then with high probability (8) and (10) are equivalent to

$$\hat{T}^{(i)} X_b^{(i)} = \hat{T}^{(i)}(F^X + X_a^{(i)} F^Z). \quad (28)$$

Corollary 2: If $\sum_{j=1}^{i} M_j - \sum_{j=1}^{i} z_j \geq b$, then the matrix equation (17) holds with high probability.

Proof: Notice that (17) are equivalent to the following set of three matrix equations:

$$B_{top} \begin{bmatrix} \boldsymbol{x}_a^{(i)} \\ \boldsymbol{x}_b^{(i)} \end{bmatrix} = \hat{\boldsymbol{T}}^{(i)} \boldsymbol{f}^X \quad (29)$$

$$B'_{mid} \begin{bmatrix} \boldsymbol{l}_a^{(i)} \\ \boldsymbol{l}_b^{(i)} \end{bmatrix} = \hat{\bar{\boldsymbol{T}}}^{(i)} \boldsymbol{f}^A \quad (30)$$

$$B_{bot} \begin{bmatrix} \boldsymbol{x}_a^{(i)} \\ \boldsymbol{x}_b^{(i)} \\ \boldsymbol{l}_a^{(i)} \\ \boldsymbol{l}_b^{(i)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h_1} \\ \vdots \\ \boldsymbol{h_i} \end{bmatrix} \quad (31)$$

But now note that (29) is equivalent to (28); (31) is equivalent to (7); and (30) is equivalent to (20) because all the deleted columns correspond to the zero padding in $L^{(i)}$. ■

Finally we need to prove that (17) has a unique solution, $i.e.$, the probability of decoding an error packet is vanishing:

Lemma 9: If $\sigma m \geq 2b\bar{c} + 2\sigma\bar{c} + 1$, then with high probability there does not exist $X' \neq X_0$ such that X' satisfies (17).

Proof: Suppose $X' \neq X_0$, and let $\boldsymbol{x}_a'$, $\boldsymbol{x}_b'$ be its vectorized versions as described in (12). We consider the probability that there exist $\boldsymbol{x}_a'$, $\boldsymbol{x}_b'$, $\boldsymbol{l}_a'$ and $\boldsymbol{l}_b'$ that satisfy (17). Let us first consider the top $\beta r_i + \gamma \bar{r}_i$ rows in B that correspond to the blocks of B_{top} and B'_{mid}

$$\begin{bmatrix} B_{top} & \boldsymbol{0} \\ \boldsymbol{0} & B'_{mid} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_a'^{(i)} \\ \boldsymbol{x}_b'^{(i)} \\ \boldsymbol{l}_a'^{(i)} \\ \boldsymbol{l}_b'^{(i)} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{T}}^{(i)} \boldsymbol{f}^X \\ \hat{\bar{\boldsymbol{T}}}^{(i)} \boldsymbol{f}^A \end{bmatrix}, \quad (32)$$

They are equivalent to

$$X_b'^{(i)} = F^X + X_a'^{(i)} F^Z \quad (33)$$
$$L_b'^{(i)} = F^A + L_a'^{(i)} F^E \quad (34)$$

Therefore given arbitrary values of $\boldsymbol{x}_a'^{(i)}$ and $\boldsymbol{l}_a'^{(i)}$, there are unique corresponding values of $\boldsymbol{x}_b'^{(i)}$ and $\boldsymbol{l}_b'^{(i)}$ that satisfy (32).

Now given any $\boldsymbol{x}_a'^{(i)}$ and $\boldsymbol{l}_a'^{(i)}$ (and the corresponding $\boldsymbol{x}_b'^{(i)}$ and $\boldsymbol{l}_b'^{(i)}$) such that (32) holds, we consider the probability that the bottom $\sum_{k=1}^{i} \alpha_k = (i^2 + i)\sigma m/2$ rows in (17) also holds:

$$B_{bot} \begin{bmatrix} \boldsymbol{x}_a'^{(i)} \\ \boldsymbol{x}_b'^{(i)} \\ \boldsymbol{l}_a'^{(i)} \\ \boldsymbol{l}_b'^{(i)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{h_1} \\ \vdots \\ \boldsymbol{h_i} \end{bmatrix} \quad (35)$$

This is equivalent to:

$$B_{bot} \begin{bmatrix} \boldsymbol{x}_a^{(i)} - \boldsymbol{x}_a'^{(i)} \\ \boldsymbol{x}_b^{(i)} - \boldsymbol{x}_b'^{(i)} \\ \boldsymbol{l}_a^{(i)} - \boldsymbol{l}_a'^{(i)} \\ \boldsymbol{l}_b^{(i)} - \boldsymbol{l}_b'^{(i)} \end{bmatrix} = \boldsymbol{0}, \quad (36)$$

Because $X' \neq X_0$, so $\boldsymbol{x}_a^{(i)} - \boldsymbol{x}_a'^{(i)}$ and $\boldsymbol{x}_b^{(i)} - \boldsymbol{x}_b'^{(i)}$ cannot both be the zero vector. Denote

$$\boldsymbol{x}_a^{(i)} - \boldsymbol{x}_a'^{(i)} = (x_{a,1}^{(i)}, ..., x_{a,\theta_a}^{(i)})^T$$
$$\boldsymbol{x}_b^{(i)} - \boldsymbol{x}_b'^{(i)} = (x_{b,1}^{(i)}, ..., x_{b,\theta_b}^{(i)})^T$$
$$\begin{bmatrix} \boldsymbol{l}_a^{(i)} - \boldsymbol{l}_a'^{(i)} \\ \boldsymbol{l}_b^{(i)} - \boldsymbol{l}_b'^{(i)} \end{bmatrix} = (l_1^{(i)}, ..., l_{\theta_l}^{(i)})^T$$

where $\theta_a = b(r_i - b)$, $\theta_b = \beta b$ and $\theta_l = (i^2 + i)\sigma m/2$. Denote the (u, v) entry of B_{bot} as $s_{u,v}$, then the j-th row of (36) is

$$\sum_{k=1}^{b(r_i-b)} x_{a,k}^{(i)} s_{j,k} + \sum_{k=1}^{\beta b} x_{b,k}^{(i)} s_{j,k+b(r_i-b)}$$
$$+ \sum_{k=1}^{(i^2+i)\sigma m/2} l_k^{(i)} s_{j,k+nb} = 0 \quad (37)$$

Let s_j be the $(j,1)$ entry of B_{bot} before column permutation, then $s_{j,k} = s_j^{\pi(k)}$, $1 \le k \le nb$, where π is a permutation of $\{1,...,nb\}$. So (37) is a non-zero polynomial of order at most $b(r_i - b) + \beta b = nb$ in variable s_j (the $\{s_{j,k+nb}\}$ are constants 0 or 1 by construction and are independent with respect to s_j). By the fundamental theorem of algebra the polynomial have at most nb roots. And the probability that s_j is chosen as one of the roots is at most nb/q, and this is the upper bound of the probability that row j holds in (36). Because $\{s_j\}$ are chosen independently, (36) holds with probability no larger than $(nb/q)^{(i^2+i)\sigma m/2}$.

Finally, there are at most $q^{b(r_i-b)}$ different $\boldsymbol{x}_a^{(i)}$ and at most $q^{i\sigma(\bar{r}_i - i\sigma)}$ different $\boldsymbol{l}_a^{(i)}$. By (8), $r_i - b \le i\bar{c}$, and by (9), $\bar{r}_i - i\sigma \le i\bar{c}$. Therefore by the union bound, the probability that there exists $X_0' \ne X_0$ such that $\boldsymbol{x}_a', \boldsymbol{x}_b', \boldsymbol{l}_a'$ and $\boldsymbol{l}_b'$ satisfy (17) is at most

$$\left(\frac{nb}{q}\right)^{\frac{(i^2+i)\sigma m}{2}} q^{ib\bar{c}+i^2\sigma\bar{c}} \le \frac{(nb)^{i^2\sigma m}}{q^{i^2}} \to 0$$

∎

We are ready to present the final conclusion.

Theorem 3: $\forall i$ such that $b + \sum_{j=1}^{i} z_j \le \sum_{j=1}^{i} M_j$, with the proposed coding scheme, Bob is able to decode X_0 correctly with high probability at the i-th stage. Otherwise, Bob waits for more redundancy instead of decoding erroneous packets.

Proof: By Corollary 2, X_0 can be solved from (17) if $b + \sum_{j=1}^{i} z_j \le \sum_{j=1}^{i} M_j$. By Lemma 9, if a solution exists, it is correct and unique. Otherwise, there is no solution to (17) and by the algorithm Bob waits for more redundancy. ∎

Similar to the case of the secret channel model, Theorem 3 shows that our code is optimal in that sense that decoding succeeds with high probability whenever the total amount of information received by the sink satisfies the cut set bound with respect to the amount of message and error information. We can also show rate-optimality under the i.i.d. case.

Theorem 4: Assume M_i, z_i, $i = 1, 2...$ are i.i.d. random variables with mean $\mathbb{E}[M]$ and $\mathbb{E}[z]$, respectively. If there exists $\epsilon > 0$ such that $\mathbb{E}[M] - \mathbb{E}[z] \ge \epsilon$, then with the proposed coding scheme Bob is able to decode X_0 correctly with high probability. And on average the code achieves rate

$$r \ge \frac{b}{b + \bar{c} - 1} \left(\mathbb{E}[M] - \mathbb{E}[z] \right).$$

Proof: Note that both long packets and short packets are sent over the network. We consider the short packets to be overhead. At the i-th stage, the length of a short packet is $i(m + \sigma)$, and is negligible as a large enough n is chosen. The rest of the proof is identical to the proof of Theorem 2. ∎

Again the proposed scheme is asymptotically rate optimal as we choose a large enough b. The computational cost of design, encoding, and decoding is dominated by the cost of solving (17), which equals $O((ni\bar{c})^3)$.

V. CONCLUSION

This paper introduces information-theoretical rateless resilient network codes against Byzantine adversaries. Unlike previous works, knowledge about the network and adversaries are not required and the codes will adapt to their parameters by sending more redundancy over time if necessary. We present two algorithms targeting two network models. The first model assumes there is a low-rate secret channel between the source and the destination. The second model assumes the source and destination share some "small" random secrets that are independent with the input information. For both models our codes are rate-optimal, distributed, polynomial-time, work on general topology, and only require source and destination nodes to be modified.

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
[2] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.
[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. on Networking*, vol. 11, no. 5, pp. 782–795, 2003.
[4] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE ISIT*, 2003.
[5] R. W. Yeung and N. Cai, "Network error correction, part i: Basic concepts and upper bounds," *Communications in Information and Systems*, vol. 6, no. 1, pp. 19 –36, 2006.
[6] N. Cai and R. W. Yeung, "Network error correction, part ii: Lower bounds," *Communications in Information and Systems*, vol. 6, no. 1, pp. 37 – 54, 2006.
[7] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Info. Theory*, August 2008.
[8] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient network coding in the presence of byzantine adversaries," *IEEE Trans. on Information Theory*, vol. 54, no. 6, pp. 2596–2603, 2008.
[9] L. Nutman and M. Langberg, "Adversarial models and resilient schemes for network coding," in *IEEE ISIT*, 2008, pp. 171–175.
[10] M. N. Krohn, M. J. Freedman, and D. Mazieres, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *IEEE Symp. Security and Privacy*, 2004, pp. 226–240.
[11] C. Gkantsidis and P. Rodriguez Rodriguez, "Cooperative security for network coding file distribution," in *IEEE INFOCOM*, 2006, pp. 1–13.
[12] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," in *Proc. 40th Annual Conf. Information Sciences and Systems*, 2006, pp. 857–863.
[13] S. Vyetrenko, A. Khosla, and T. Ho, "On combining information-theoretic and cryptographic approaches to network coding security against the pollution attack," in *Proc. Conf Signals, Systems and Computers Record of the Forty-Third Asilomar Conf*, 2009, pp. 788–792.
[14] M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudo random bits," in *Proc. 23rd Annual Symp. Foundations of Computer Science*, 1982, pp. 112–117.
[15] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: a decentralized approach based on network coding," in *IEEE INFOCOM*, vol. 3, 2005, pp. 1607–1617.
[16] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. on Info. Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
[17] A. W. Naylor and G. R. Sell, *Linear operator theory in engineering and science*. Springer, 1994.
[18] T. Ho, M. Mdard, J. Shi, M. Effros, and D. Karger, "On randomized network coding," in *in Proc. 41st Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, 2003.
[19] G. R. Grimmett and D. R. Stirzaker, *Probability and random processes*. Oxford university press, 2001.
[20] S. Jaggi, M. Langberg, T. Ho, and M. Effros, "Correction of adversarial errors in networks," in *IEEE ISIT*, 2005, pp. 1455–1459.